ELSEVIER

Theoretical Computer Science 207 (1998) 171–180

Theoretical Computer Science

Approximations for subset interconnection designs

Xiufeng Du [a], Weili Wu [b,*,1], Dean F. Kelley [b,1]

[a] *Department of Mathematics, Qiqihar Teacher's College, Minneapolis, MN 55455, USA*
[b] *Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA*

Abstract

Given a complete weighted graph on vertex set X and subsets $X_1, \dots, X_m$ of X, we consider the problem of finding a minimum total weight subgraph G such that for every $i = 1, \dots, m$, G contains a spanning tree for X_i. The NP-hardness of this problem was established in 1985 under Ronald V. Book's supervision. In this note, we present some results about its polynomial-time approximation. © 1998 Published by Elsevier Science B.V. All rights reserved

Keywords: Subset interconnection; Approximation; Performance ratio

1. Introduction

Given a complete weighted graph on the set X of n vertices and subsets $X_1, \dots, X_m$ of X, we consider the problem of finding a minimum total weight subgraph G such that for every $i = 1, \dots, m$, G contains a spanning tree for X_i. We will refer this problem as the SID (subset interconnection designs). The SID has applications in computer science and statistics [8, 9, 1] and has been studied extensively in the literature [2, 4–7, 11, 12]. The NP-hardness of the SID was first proved in a Ph.D. thesis [3] under Ronald V. Book's supervision. Prisner [11] introduced a polynomial-time heuristic with performance ratio $\ln m + O(1)$, that is, the heuristic produces an approximation within a factor of $\ln m + O(1)$ from the optimal. However, his proof is incorrect. In this note, we will give a correct proof. In addition, we show that the SID has no polynomial-time approximation with performance ratio $\rho \log m$ for $0 < \rho < 1/4$ unless $NP \subset DTIME(n^{poly \log n})$. This means that Prisner's heuristic has almost best possible performance ratio. We also present another heuristic with performance ratio depending on only the maximum size of X_i's.

Let $c(\cdot)$ be a weight function on the set of edges between vertices in X. A graph G with vertex X is called a *feasible graph* for $(X_1, X_2, \dots, X_m)$ if for any $i = 1, 2, \dots, m$,

* Corresponding author. E-mail: wuw@cs.umn.edu.

[1] Supported in part by the NSF under grants CCR-9208913 and CCR-9530306.

PII S0304-3975(98)00062-0

the subgraph $G[X_i]$ induced by X_i is connected. A feasible graph is *minimum* if G is an optimal solution for the SID with weight function $c(\cdot)$. For a graph G, we denote by $c(G)$ the total weight of edges in G. For any set Y, we denote by $|Y|$ the number of elements in Y. For any edge e, we denote by $I(e)$ the set of indices i such that X_i contains both endpoints of e.

2. Prisner's approximation

If the two endpoints of an edge belong to more X_i's, then the edge is more likely to appear in a minimum feasible graph. From this observation, Prisner [11] proposed the concept of "benefit" as follows.

For a graph G and a set system $(X_1, \ldots, X_m)$, define the benefit function by

$$u(e, G) = \sum_{i=1}^{m} u_i(e, G),$$

where

$$u_i(e, G) = \begin{cases} 1 & \text{if the edge } e \text{ connects two connected components of } G[X_i], \\ 0 & \text{otherwise.} \end{cases}$$

The benefit-cost ratio of e to G is $u(e, G)/c(e)$ where $c(e)$ is the weight of the edge e. Using this concept, Prisner [11] discovered a $\sum_{i=1}^{K} \frac{1}{i}$-heuristic for the SID where K is the maximum number of indices belonging to an edge, that is, $K = \max_{x,y \in X} |I(x, y)|$. His approximation algorithm runs in time $O(n^4 + mn^2)$ as follows.

Algorithm P

```
begin
    G := ∅;
    while there is an edge of positive benefit do
    choose an edge e with the largest benefit-cost ratio u(e,G)/c(e) and
      set G := G ∪ e.
end.
```

Prisner [11] also proved the following.

Theorem 2.1. *Algorithm P produces an approximation solution within a factor of* $\sum_{i=1}^{K} 1/i$ *from the optimal.*

However, his proof is incorrect. To see this, let us follow his notation and argument as follows.

Let G be the graph obtained by Algorithm P. Let $e_1, e_2, \ldots, e_r$ be the edges of G in the order of their appearance. For every $1 \leqslant t \leqslant r$, denote by G_t the graph with vertex set X and edge set $\{e_1, e_2, \ldots, e_t\}$. Define

$$I_t = \{i \in I(e_t) \mid u_i(e_t, G_{t-1}) = 1\}.$$

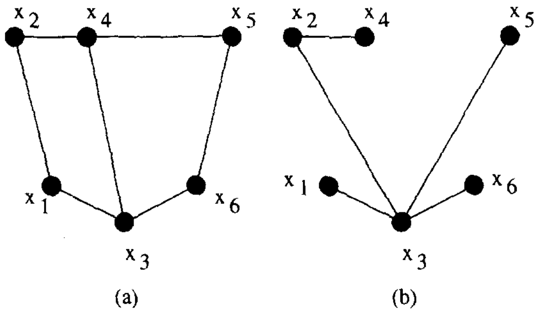

Fig. 1. Counterexample.

Let G^* be the minimum feasible graph. Let $e_1^*, e_2^*, \ldots, e_s^*$ the edges of G^* in an arbitrary order. For every $1 \leqslant j \leqslant s$, denote by G_j^* the graph with vertex set X and edge set $\{e_1^*, e_2^*, \ldots, e_j^*\}$. Define

$$I_j^* = \{i \in I(e_j^*) \mid u_i(e_j^*, G_{j-1}^*) = 1\}.$$

For each $j = 1, 2, \ldots, s$, the two endpoints of e_j^* are connected in $G[X_i]$ for every $i \in I(e_j^*)$. Thus, for each i, there is the smallest integer p_i for which they are already connected in $G_{p_i}[X_i]$. Let $l(j,1) \leqslant l(j,2) \leqslant \cdots \leqslant l(j, |I_j^*|)$ denote these p_i's.

Prisner proved that "for any $1 \leqslant t \leqslant r$ there are exactly $|I_t|$ airs (j,q) with $1 \leqslant j \leqslant s$, $1 \leqslant q \leqslant |I_j^*|$, and $l(j,q) = t$." This fact then becomes a crucial tool to prove the theorem. However, this fact is incorrect. The following is a counterexample.

Counterexample. Consider six points x_i for $i = 1, \ldots, 6$. Define

$$X_1 = \{x_1, x_2, x_3\}, \qquad X_2 = \{x_2, x_3, x_4, x_5\}, \qquad X_3 = \{x_3, x_5, x_6\}$$

and assign weights by

$$c(x_1, x_3) = c(x_3, x_6) = c(x_2, x_4) = \varepsilon,$$

$$c(x_1, x_2) = 1 - \varepsilon, \qquad c(x_5, x_6) = 1,$$

$$c(x_4, x_5) = 1 + 2\varepsilon, \qquad c(x_3, x_4) = 1 + 3\varepsilon,$$

$$c(x_3, x_2) = c(x_3, x_5) = 2,$$

$$c(x_i, x_j) = 6 \text{ for other edges } (x_i, x_j),$$

where ε is a sufficiently small positive number.

The feasible graph G obtained from Algorithm P is shown in Fig. 1 (a) and the minimum feasible graph G^* is shown in Fig. 1(b). The edges of G in the order of their appearance are $e_1 = (x_1, x_3)$, $e_2 = (x_3, x_6)$, $e_3 = (x_2, x_4)$, $e_4 = (x_1, x_2)$, $e_5 = (x_5, x_6)$, $e_6 = (x_4, x_5)$, $e_7 = (x_3, x_4)$. The edges of G^* are $e_1^* = e_1$, $e_2^* = e_2$, $e_3^* = e_3$, $e_4^* =$

(x_3,x_2), $e_5^* = (x_3,x_5)$. Clearly, $I_1 = \{1\}$, $I_2 = \{3\}$, $I_3 = \{2\}$, $I_4 = \{1\}$, $I_5 = \{3\}$, $I_6 = \{2\}$, $I_7 = \{2\}$ and $I_1^* = \{1\}$, $I_2^* = \{3\}$, $I_3^* = \{2\}$, $I_4^* = \{1,2\}$, $I_5^* = \{2,3\}$. Thus, $|I_6| = |I_7| = 1$. However, for each edge e_j^*, the integers $l(j,1) \leqslant l(j,2) \leqslant \cdots \leqslant l(j,|I_j^*|)$ are as follows:

For $e_1^*, l(1,1) = 1$. For $e_2^*, l(2,1) = 2$. For $e_3^*, l(3,1) = 3$.

For $e_4^*, l(4,1) = 4$ and $l(4,2) = 7$. For $e_5^*, l(5,1) = 5$ and $l(5,2) = 7$.

Therefore, for $t = 6$, no pair (j,q) exists with $1 \leqslant j \leqslant s$, $1 \leqslant q \leqslant |I_j^*|$, and $l(j,q) = 6$, and for $t = 7$, there are two pairs (j,q) with $1 \leqslant j \leqslant s$, $1 \leqslant q \leqslant |I_j^*|$, and $l(j,q) = 7$. (These two pairs are $(4,2)$ and $(5,2)$.)

In the following, we give a correct proof.

Proof of Theorem 2.1. Let G be the approximation solution produced by Algorithm P and $e_1, e_2, \ldots, e_g$ all edges of G in the order of appearance in the algorithm. For every $t = 1,2,\ldots,g$, denote by G_t the graph with vertex set V and edge set $\{e_1, e_2, \ldots, e_t\}$. By the greedy rule, we have

$$\frac{u(e_1,G)}{c(e_1)} \geqslant \frac{u(e_2,G_1)}{c(e_2)} \geqslant \cdots \geqslant \frac{u(e_g,G_{g-1})}{c(e_g)}.$$

We define a weight function w as follows: For every e_t and every index $i \in I(e_t)$, with $u_i(e_t,G_{t-1}) = 1$, we define

$$w(e_t,i) = \begin{cases} c(e_t)/u(e_t,G_{t-1}) & \text{if } u_i(e_t,G_{t-1}) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose G^* is a minimum feasible graph. For every edge e in G^* and every index $i \in I(e)$, define $w^*(e,i) = w(e_t,i)$ if t is the smallest integer such that the two endpoints of e are already connected in $G_t[X_i]$.

Our first claim is that

$$\sum_{i \in I(e)} w^*(e,i) \leqslant (1 + \log_2 K) c(e).$$

To show this claim, suppose that $I(e) = \{1', 2', \ldots, k'\}$ and $w^*(e,i') = w(e_{i''},i')$ (i.e., i' is the smallest index such that the two endpoints of e are already connected in $G_{i'}[X_{i''}]$). Without loss of generality, assume $1'' \geqslant 2'' \geqslant \cdots \geqslant k''$. Then we must have $u(e, G_{i''-1}) \geqslant i$. By the greedy rule, we have

$$\frac{u(e_{i''},G_{i''-1})}{c(e_{i''})} \geqslant \frac{u(e,G_{i''-1})}{c(e)} \geqslant \frac{i}{c(e)}.$$

Therefore,

$$\begin{aligned} \sum_{i \in I(e)} w^*(e,i) &= \sum_{i=1}^{k} w^*(e,i') = \sum_{i=1}^{k} w(e_{i''},i') = \sum_{i=1}^{k} \frac{c(e_{i''})}{u(e_{i''},G_{i''-1})} \\ &\leqslant \sum_{i=1}^{k} \frac{c(e)}{i} \leqslant (1 + \log_2 K) c(e). \end{aligned}$$

Our second claim is that for any $i = 1, 2, \ldots, m$,

$$\sum_{1 \leqslant t \leqslant g, u_i(e_t, G_{t-1})=1} w(e_t, i) \leqslant \sum_{e \in E(G^*[X_i])} w^*(e, i),$$

where $E(G^*)$ denotes the edge set of a graph G^*. To see this, we note that all edges e_t with $u_i(e_t, G_{t-1}) = 1$ form a spanning tree for s_i. Suppose that those edges are $e_{1'}, e_{2'}, \ldots, e_{h'}$ with $1' < 2' < \cdots < h'$ and $h = |X_i| - 1$. Then $G_{j'-1}[X_i]$ $(1 \leqslant j \leqslant h)$ must have exactly $h - j + 1$ connected component. This means that there are at least $h - j$ edges in $G^*[X_i]$ whose endpoints have not been connected in $G_{j'-1}$. For each of those edges e in $G[X_i]$, we have $w^*(e,i) \geqslant w(e_{j'}, i)$. Note that $w(e_{1'}, i) \leqslant w(e_{2'}, i) \leqslant \cdots \leqslant w(e_{h'}, i)$. Moreover, we already proved that $G^*[X_i]$ has at least one edge e with $w^*(e,i) \geqslant w(e_{h'}, i)$, at least two edges e's with $w^*(e,i) \geqslant w(e_{(h-1)'}, i), \ldots$, at least h edges e's with $w^*(e,i) \geqslant w(e_{1'}, i)$. Therefore, $\sum_{e \in E(G^*[X_i])} w^*(e,i) \geqslant \sum_{j=1}^{h} w(e_{j'}, i)$.

Now, by our two claims, we have

$$\begin{aligned} c(G) &= \sum_{t=1}^{g} \sum_{i \in I(e_t), u_i(e_t, G_{t-1})=1} w(e_t, i) = \sum_{i=1}^{m} \sum_{1 \leqslant t \leqslant g, u_i(e_t, G_{t-1})=1} w(e_t, i) \\ &\leqslant \sum_{i=1}^{m} \sum_{e \in E(G^*[s_i])} w^*(e,i) = \sum_{e \in E(G^*)} \sum_{i \in I(e)} w^*(e,i) \leqslant \sum_{e \in E(G^*)} (1 + \log_2 K) c(e) \\ &= (1 + \log_2 K) c(G^*). \qquad \square \end{aligned}$$

Next, we show a lower bound for the performance ratio of any polynomial-time approximation for SID.

Theorem 2.2. *For $0 < \rho < 1/4$, the SID has no polynomial-time heuristic which produces an approximation solution within a factor of $\rho \log m$ from the optimal unless $NP \subset DTIME(n^{poly \log n})$.*

The proof is based on a recent result of Lund and Yannakakis [10]. Consider the following problem.

Set Covering: Given a (finite) collection $\mathscr{S}$ of subsets of a set U of m elements, find a minimum cardinality subcollection of $\mathscr{S}$ such that the union of all subsets in the subcollection covers U.

They proved the following.

Theorem 2.3 (Lund and Yannakakis [10]). *For any $0 < \rho < 1/4$, there is no polynomial-time approximation algorithm with performance ratio $\rho \log m$ for Set Covering problem unless $NP \subset DTIME\ (n^{poly \log n})$.*

Now, we use their result to prove our theorem.

Proof of Theorem 2.3. We reduce Set Covering to SID as follows. For each instance $\mathscr{S}$ of Set Covering, assume that $U = \{1,2,\dots,m\}$ and $\mathscr{S} = \{S_1,\dots,S_n\}$. We construct an instance of SID by setting

$$X_i = \{j \mid i \in S_j\} \cup \{0\}$$

and assigning weights as follows:

$$c(j,k) = \varepsilon, \quad 1 \leqslant j < k \leqslant n,$$
$$c(0,j) = 1, \quad 1 \leqslant j \leqslant n,$$

where $\varepsilon \leqslant \frac{1}{n(n-1)}$. For each set cover $\{S_{j_1},\dots,S_{j_s}\}$, we can construct a feasible graph by connecting edges $(0,j_1),\dots,(0,j_s)$ to the complete graph on $\{1,\dots,n\}$.

Clearly, if the optimal solution for Set Covering is $\mathscr{S}_{opt}$, a subcollection of s sets, then the minimum feasible graph for $(X_1,\dots,X_m)$ has total weight between s and $s + \frac{1}{2}n(n-1)\varepsilon$.

Suppose to the contrary that there exists a polynomial-time approximation algorithm with performance ratio $\rho \log m$ for SID where $0 < \rho < 1/4$. Let G be the feasible graph obtained by this approximation algorithm. Define

$$\mathscr{S}' = \{S_j \mid \text{edge } (0,j) \text{ is in } G\}.$$

Clearly, $\mathscr{S}'$ is a set cover of U and

$$c(G) - \tfrac{1}{2}n(n-1)\varepsilon \leqslant |\mathscr{S}'| \leqslant c(G).$$

Therefore,

$$\begin{aligned} |\mathscr{S}'| &\leqslant c(G) \\ &\leqslant \rho \log m \cdot \left(s + \frac{n(n-1)}{2}\varepsilon\right) \\ &\leqslant \left(\rho + \frac{1}{2s}\right) \log m \cdot |\mathscr{S}_{opt}|. \end{aligned}$$

Now, we compute an approximation solution for Set Covering as follows:

Step 1: Check all subcollections of size at most $1/(0.25 - \rho)$. If there exists a set cover of U among them, then choose the one with minimum cardinality as solution; else, go to step 2.

Step 2: Find an approximation solution through the above reduction and the polynomial-time approximation algorithm for SID.

Clearly, if the solution comes from Step 1, then it is optimal; if the solution comes from Step 2, then we must have $|\mathscr{S}_{opt}| = s \geqslant 1/(0.25-\rho)$. Therefore, this approximation solution is within a factor of $(0.25 + \rho)/2 \cdot \log m$ from the optimal. By Theorem 2.4 NP $\subset$ DTIME $(n^{poly \log n})$. $\square$

3. Approximations for small size X_i's

It is interesting to point out that there exist heuristics with performance ratio not depending on m. Suppose $|X_i| \leqslant \eta$ for every $i = 1, \ldots, m$. We present heuristics depending on only η in this section.

Suppose $X = \{1, \ldots, n\}$. Let x_{ij} be a variable representing edge (i,j) in the way that if $x_{ij} = 1$, then edge (i,j) exsits and if $x_{ij} = 0$, then edge (i,j) does not exist. Then any graph on X can be given by an assignment to all variables x_{ij} for $i,j \in X$. That the subgraph induced by X_i is connected is equivalent to that the assignment satisfies the following system of inequalities:

$$\sum_{i \in A} \sum_{j \in B} x_{ij} \geqslant 1$$

for all partition (A,B) of X_i. ((A,B) is a partition of X_i if $A \cup B = X_i$ and $A \cap B = \emptyset$.) Therefore, the SID is equivalent to the following problem.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{1 \leqslant i < j \leqslant n} c_{ij} x_{ij} \\ \text{subject to} & \displaystyle\sum_{i \in A} \sum_{j \in B} x_{ij} \geqslant 1 \quad \text{for all } k \in \{1, \ldots, m\} \text{ and all partition } (A,B) \text{ of } X_k \\ & x_{ij} = 0 \text{ or } 1 \quad \text{for } 1 \leqslant i < j \leqslant n \end{array}$$

where $c_{ij} = c(i,j)$. Assume that $|X_k| \leqslant \eta$ for all $k = 1, \ldots, m$. Define $\tau = \lfloor \eta^2/4 \rfloor$. Based on the above formulation, we can give the following heuristic.

Algorithm L1

Step 1: Solve the following linear programming.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{1 \leqslant i < j \leqslant n} c_{ij} x_{ij} \\ \text{subject to} & \displaystyle\sum_{i \in A} \sum_{j \in B} x_{ij} \geqslant 1 \quad \text{for all } k \in \{1, \ldots, m\} \text{ and all partition } (A,B) \text{ of } X_k \\ & 0 \leqslant x_{ij} \leqslant 1 \quad \text{for } 1 \leqslant i < j \leqslant n. \end{array}$$

Suppose that the solution is $(x^*_{ij})_{1 \leqslant i < j \leqslant n}$.

Step 2: Set

$$x'_{ij} = \begin{cases} 1 & \text{if } x^*_{ij} \geqslant 1/\tau, \\ 0 & \text{otherwise.} \end{cases}$$

Take $(x'_{ij})_{1 \leqslant i < j \leqslant n}$ to be an approximation solution for the SID.

Theorem 3.1. *For fixed η, Algorithm L1 runs in polynomial time and produces an approximation solution for the SID within a factor of τ from the optimal.*

Proof. The number of constraints in the linear programming is $O(m2^\eta)$. Thus, for fixed η, the algorithm runs in polynomial time. Now, we show that $(x'_{ij})_{1\leqslant i<j\leqslant n}$ is a feasible solution for the 0–1 programming equivalent to the SID. Note that for any $k = 1,\dots,m$ and any partition (A,B) of X_k,

$$|A|\cdot|B| \leqslant \left(\frac{|X_k|}{2}\right)^2 \leqslant \left(\frac{\eta}{2}\right)^2.$$

Since $|A|\cdot|B|$ is an integer, we have $|A|\cdot|B|\leqslant\tau$. Thus, from $\sum_{i\in A}\sum_{j\in B} x^*_{ij}\geqslant 1$, we know that at least one x^*_{ij} in this sum is not smaller than $1/\tau$. It follows from the definition of x'_{ij} that $\sum_{i\in A}\sum_{j\in B} x'_{ij}\geqslant 1$. This means that $(x'_{ij})_{1\leqslant i<j\leqslant n}$ is a feasible solution for the 0-1 programming. Let G' be the feasible graph consisting of all edges (i,j) for $x'_{ij} = 1$. Then $c(G') = \sum_{1\leqslant i<j\leqslant n} x'_{ij}$. Let G^* be a minimum feasible graph for SID. Then $c(G^*)\geqslant \sum_{1\leqslant i<j\leqslant n} c_{ij}x^*_{ij}$. Thus

$$c(G')\leqslant \sum_{1\leqslant i<j\leqslant n} c_{ij}x'_{ij} \leqslant \tau \sum_{1\leqslant i<j\leqslant n} c_{ij}x^*_{ij} \leqslant \tau c(G^*). \qquad \square$$

Du and Miller [6] showed that the SID with unit weight function in the case that $|X_k|\leqslant 3$ for all $k = 1,\dots,m$ is still NP-hard and has a polynomial-time approximation within a factor of 2 from the optimal. The following is an extension of their result from unit weight function to the general weight function.

Corollary 3.2. *If $|X_i|\leqslant 3$ for all $i = 1,\dots,m$, then there exists a polynomial-time approximation for SID within a factor of 2 from the optimal.*

If $|X_i|\leqslant 4$ for all $i = 1,\dots,m$, then by Theorem 3.1, there exists a polynomial-time approximation for SID within a factor of 4 from the optimal. Next, we provide a better result.

First, note that a graph on $h\leqslant 4$ vertices is connected if and only if the graph contains at least $h-1$ edges and every vertex is incident to at least one edge. Thus, in the case that $|X_i|\leqslant 4$ for all $i = 1,\dots,m$, the SID is equivalent to the following.

$$\begin{aligned}
\text{minimize} \quad & \sum_{1\leqslant i<j\leqslant n} c_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{j\in X_k\setminus\{i\}} x_{ij}\geqslant 1 \quad \text{for all } i\in X_k \text{ and all } k = 1,\dots,m \\
& \sum_{i,j\in X_k, i\neq j} x_{ij}\geqslant |X_k| - 1 \quad \text{for all } k = 1,\dots,m \\
& x_{ij} = 0 \text{ or } 1 \quad \text{for } 1\leqslant i < j\leqslant n
\end{aligned}$$

Based on this formulation, we can give the following heuristic.

Algorithm L2

Step 1: Solve the following linear programming.

$$\begin{aligned}
\text{minimize} \quad & \sum_{1\leqslant i<j\leqslant n} c_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{j\in X_k\setminus\{i\}} x_{ij} \geqslant 1 \quad \text{for all } i\in X_k \text{ and all } k=1,\ldots,m \\
& \sum_{i,j\in X_k, i\neq j} x_{ij} \geqslant |X_k|-1 \quad \text{for all } k=1,\ldots,m \\
& 0\leqslant x_{ij}\leqslant 1 \quad \text{for } 1\leqslant i<j\leqslant n
\end{aligned}$$

Suppose that the solution is $(x^*_{ij})_{1\leqslant i<j\leqslant n}$.

Step 2: Set

$$x'_{ij} = \begin{cases} 1 & \text{if } x^*_{ij}\geqslant 1/3, \\ 0 & \text{otherwise.} \end{cases}$$

Take $(x'_{ij})_{1\leqslant i<j\leqslant n}$ to be an approximation solution for the SID.

Theorem 3.3. *Algorithm L2 produces an approximation solution for the SID within a factor of* 3 *from the optimal.*

Proof. First, we show that $(x'_{ij})_{1\leqslant i<j\leqslant n}$ is a feasible solution for the above 0-1 programming equivalent to the SID in the case that $|X_k|\leqslant 4$ for all $k=1,\ldots,m$.

(1) For any $k=1,\ldots,m$ and any $i\in X_k$, since $\sum_{j\in X_k\setminus\{i\}} x^*_{ij}\geqslant 1$, there exists at least one j in $X_k\setminus\{i\}$ such that $x^*_{ij}\geqslant 1/3$. This means that $\sum_{j\in X_k\setminus\{i\}} x'_{ij}\geqslant 1$.

(2) For any $k=1,\ldots,m$, since $\sum_{i,j\in X_k,i\neq j} x^*_{ij}\geqslant |X_k|-1$, there exists at least $|X_k|-1$ x^*_{ij}'s, in the sum, not smaller than $1/3$. This implies that $\sum_{i,j\in X_k,i\neq j} x'_{ij}\geqslant |X_k|-1$.

By (1) and (2), $(x'_{ij})_{1\leqslant i<j\leqslant n}$ is feasible for the 0-1 programming. Let G' be the feasible graph consisting of all edges (i,j) for $x'_{ij}=1$. Then $c(G')=\sum_{1\leqslant i<j\leqslant n} x'_{ij}$. Let G^* be a minimum feasible graph for SID. Then $c(G^*)\geqslant \sum_{1\leqslant i<j\leqslant n} c_{ij}x^*_{ij}$. Thus

$$c(G') = \sum_{1\leqslant i<j\leqslant n} c_{ij}x'_{ij} \leqslant 3 \sum_{1\leqslant i<j\leqslant n} c_{ij}x^*_{ij} \leqslant 3c(G^*). \qquad \square$$

4. Final remark

After we completed this paper, L.A. Wolsey pointed out that the problem of finding a minimum weight set that is a spanning set in each of n matroids has been studied in [13] and his result about greedy algorithm for this problem contains Prisner's theorem as a special case, however, our proof in this paper is different from his one.

Acknowledgements

We wish to thank Professor L.A. Wolsey for pointing out some of his earlier results and providing helpful references.

References

[1] C. Beeri, R. Fagin, D. Maier, M. Yannakakis, On the decidability of acyclic database schemes, J. ACM 30 (1983) 479–513.
[2] D.-Z. Du, Y.-M. Chen, Placement of valves in vacuum systems, J. Electric Light Sources 4 (1976) (in Chinese).
[3] D.-Z. Du, Generalized complexity cores and levelability of intractable sets, Ph.D. Thesis, University of California, Santa Barbara, 1985.
[4] D.-Z. Du, An optimization problem, Discrete Appl. Math. 14 (1986) 101–104.
[5] D.-Z. Du, D.F. Kelley, On complexity of subset interconnection designs, J. Global Optim., to appear.
[6] D.-Z. Du, Z. Miller, Matroids and subset interconnection design, SIAM J. Discrete Math. 1 (1988) 416–424.
[7] D.-Z. Du, P.M. Pardalos, Subset interconnection designs: generalization of spanning trees and Steiner trees, in: D.-Z. Du, P.M. Pardalos (Eds.), Network Optimization Problems, World Scientific, Singapore, 1993, pp. 53–62.
[8] P. Duchet, Propriete de Helly et problemes de representation, in: Colloque International Paris-Orsay 260 (1978) 117–118.
[9] F.K. Hwang, D.S. Richards, P. Winter, The Steiner Tree Problem, North-Holland, Amsterdam, 1992.
[10] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, Proc. 25th Annual ACM Symp. on the Theory of Computing, San Diego, 1993, pp. 286–293.
[11] E. Prisner, Two algorithms for the subset interconnection design, Networks 22 (1992) 385–395.
[12] T.-Z. Tang, An optimality condition for minimum feasible graphs, Appl. Math. 2 (1989) 21–24 (in Chinese).
[13] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem, Combinatorica 2 (4) (1982) 385–393.